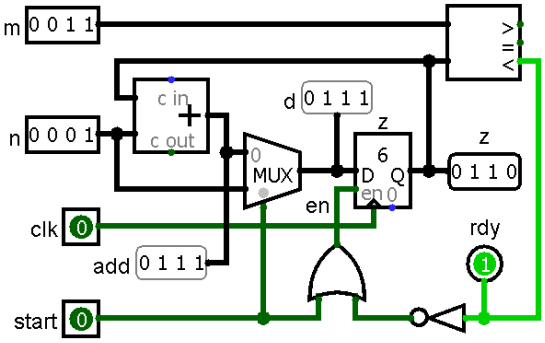
Pregunta 1

I. Considere el circuito de la figura. Los valores de *m* y *n* son constantes en 0b11 y 0b1 respectivamente. **Complete** la tabla de abajo con los valores que toma cada señal hasta el ciclo 6 del reloj. Use notación binaria.



ciclo	start	add	d	z	en	rdy
1	0	111	111	110	0	1
2	1	111	1	**110**	1	1
3	0			**1**		
4	0					
5	0					
6	0					

II. Traduzca la función *palindro* a assembler Risc-V:

```
int palindro(int *a, int n){
  for (int i=0; i<n/2; i++){
    if (a[i]!=a[n-1-i])
      return i;
  }
  return n;
}
```

Pregunta 2

Programe la función *desplazarDer* con el siguiente encabezado:

void ***desplazarDer****(char *s, int n);*

Esta función desplaza el string *s* en *n* caracteres hacia la derecha agregando espacios en blanco al principio y eliminando los últimos *n* caracteres. Está permitido usar *strlen*. Este es un ejemplo de uso:

```
char s[]= "quien vive";
desplazarDer(s, 3); // s es el string "   quien v" con
                    // 3 espacios al inicio
```

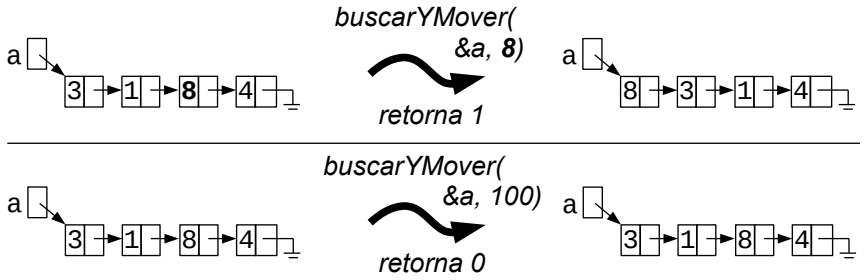
Restricciones: No use el operador de subindicación de arreglos [ ] ni su equivalente *(p+i), use aritmética de punteros. Sí puede usar *p+i*. No puede pedir memoria con *malloc* ni declarar arreglos.

Ayuda: Si *L* es el largo del string, Ud. *debe* recorrer los caracteres desde la posición *L-n-1* hacia la posición 0, copiándolos en su posición de destino. Si recorre de principio a fin sobreescribirá incorrectamente caracteres que no ha copiado todavía.

Pregunta 3

Programe la función *buscarYMover* con el encabezado del cuadro que viene a continuación. Esta función busca *z* en la lista enlazada **pa* (no ordenada, sin duplicados). En caso de éxito retorna 1 y mueve el nodo que contiene *z* al inicio de la lista enlazada. En caso contrario retorna 0. La figura de más abajo muestra 2 ejemplos de uso. El tipo de *a* es *Nodo**. En el primer ejemplo se busca 8 en la lista con éxito, retornando uno y moviendo el nodo que contiene 8 al comienzo de la lista. En el segundo ejemplo la búsqueda fracasa y por lo tanto se retorna 0 sin modificar la lista.

```
typedef struct nodo {
  int x;
  struct nodo *prox;
} Nodo;
int buscarYMover(Nodo **pa,
                 int z);
```



Restricciones: No puede modificar el campo *x* en los nodos. Debe mover el nodo completo. No puede pedir más memoria con *malloc*.

Pregunta 4

La función *primos* del cuadro de la derecha selecciona los números primos del arreglo *a* de *n* elementos, moviéndolos al inicio de *a*. Además retorna la cantidad de números primos encontrados. Este es un ejemplo de uso:

```
int a[5]= {10,3,20,5,4};
int k= primos(a, 5);
      // k es 2, a[0] es 3 y a[1] es 5
```

```
int primos(int a[]
           int n) {
  int *p= a;
  int *q= a;
  for (int i=0;i<n;i++){
    int x= *p++;
    if (esPrimo(x))
      *q++= x;
  }
  return q-a;
}
```

No interesan los valores finales de *a*[2], *a*[3] y *a*[4].

Programe la función *primosPar* con los mismos parámetros, mismo valor retornado e igual funcionalidad que *primos*, pero en paralelo para un computador dual core.

Métodología obligatoria: Use *fork* para crear un solo proceso hijo que se comunica con el proceso padre por medio de un pipe. No puede crear otros procesos. En el padre seleccione los primos de una mitad del arreglo y en el hijo los primos de la otra mitad. Use el pipe para enviar al padre la cantidad de primos encontrados por el hijo y el subarreglo con los primos seleccionados por el hijo. **Verifique que haya paralelismo** en su solución.